Postel

Network Working Group
Request for Comments: 1036
Obsoletes: RFC-850

M. Horton
AT&T Bell Laboratories
R. Adams
Center for Seismic Studies
December 1987

Standard for Interchange of USENET Messages

STATUS OF THIS MEMO

This document defines the standard format for the interchange of network News messages among USENET hosts. It updates and replaces RFC-850, reflecting version B2.11 of the News program. This memo is disributed as an RFC to make this information easily accessible to the Internet community. It does not specify an Internet standard. Distribution of this memo is unlimited.

1. Introduction

This document defines the standard format for the interchange of network News messages among USENET hosts. It describes the format for messages themselves and gives partial standards for transmission of news. The news transmission is not entirely in order to give a good deal of flexibility to the hosts to choose transmission hardware and software, to batch news, and so on.

There are five sections to this document. Section two defines the format. Section three defines the valid control messages. Section four specifies some valid transmission methods. Section five describes the overall news propagation algorithm.

2. Message Format

The primary consideration in choosing a message format is that it fit in with existing tools as well as possible. Existing tools include implementations of both mail and news. (The notesfiles system from the University of Illinois is considered a news implementation.) A standard format for mail messages has existed for many years on the Internet, and this format meets most of the needs of USENET. Since the Internet format is extensible, extensions to meet the additional needs of USENET are easily made within the Internet standard. Therefore, the rule is adopted that all USENET news messages must be formatted as valid Internet mail messages, according to the Internet standard RFC-822. The USENET News standard is more restrictive than the Internet standard,

placing additional requirements on each message and forbidding use of certain Internet features. However, it should always be possible to use a tool expecting an Internet message to process a news message. In any situation where this standard conflicts with the Internet standard, RFC-822 should be considered correct and this standard in error.

Here is an example USENET message to illustrate the fields.

```
From: jerry@eagle.ATT.COM (Jerry Schwarz)
Path: cbosgd!mhuxj!mhuxt!eagle!jerry
Newsgroups: news.announce
Subject: Usenet Etiquette -- Please Read
Message-ID: <642@eagle.ATT.COM>
Date: Fri, 19 Nov 82 16:14:55 GMT
Followup-To: news.misc
Expires: Sat, 1 Jan 83 00:00:00 -0500
Organization: AT&T Bell Laboratories, Murray Hill

The body of the message comes here, after a blank line.
```

Here is an example of a message in the old format (before the existence of this standard). It is recommended that implementations also accept messages in this format to ease upward conversion.

```
From: cbosgd!mhuxj!mhuxt!eagle!jerry (Jerry Schwarz)
Newsgroups: news.misc
Title: Usenet Etiquette -- Please Read
Article-I.D.: eagle.642
Posted: Fri Nov 19 16:14:55 1982
Received: Fri Nov 19 16:59:30 1982
Expires: Mon Jan 1 00:00:00 1990

The body of the message comes here, after a blank line.
```

Some news systems transmit news in the A format, which looks like this:

```
Aeagle.642
news.misc
cbosgd!mhuxj!mhuxt!eagle!jerry
Fri Nov 19 16:14:55 1982
Usenet Etiquette - Please Read
The body of the message comes here, with no blank line.
```

A standard USENET message consists of several header lines, followed by a blank line, followed by the body of the message. Each header

line consist of a keyword, a colon, a blank, and some additional information. This is a subset of the Internet standard, simplified to allow simpler software to handle it. The "From" line may optionally include a full name, in the format above, or use the Internet angle bracket syntax. To keep the implementations simple, other formats (for example, with part of the machine address after the close parenthesis) are not allowed. The Internet convention of continuation header lines (beginning with a blank or tab) is allowed.

Certain headers are required, and certain other headers are optional. Any unrecognized headers are allowed, and will be passed through unchanged. The required header lines are "From", "Date", "Newsgroups", "Subject", "Message-ID", and "Path". The optional header lines are "Followup-To", "Expires", "Reply-To", "Sender", "References", "Control", "Distribution", "Keywords", "Summary", "Approved", "Lines", "Xref", and "Organization". Each of these header lines will be described below.

2.1. Required Header lines

2.1.1. From

The "From" line contains the electronic mailing address of the person who sent the message, in the Internet syntax. It may optionally also contain the full name of the person, in parentheses, after the electronic address. The electronic address is the same as the entity responsible for originating the message, unless the "Sender" header is present, in which case the "From" header might not be verified. Note that in all host and domain names, upper and lower case are considered the same, thus "mark@cbosgd.ATT.COM", "mark@cbosgd.att.com", and "mark@CBosgD.ATt.COm" are all equivalent. User names may or may not be case sensitive, for example, "Billy@cbosgd.ATT.COM" might be different from "BillY@cbosgd.ATT.COM". Programs should avoid changing the case of electronic addresses when forwarding news or mail.

RFC-822 specifies that all text in parentheses is to be interpreted as a comment. It is common in Internet mail to place the full name of the user in a comment at the end of the "From" line. This standard specifies a more rigid syntax. The full name is not considered a comment, but an optional part of the header line. Either the full name is omitted, or it appears in parentheses after the electronic address of the person posting the message, or it appears before an electronic address which is enclosed in angle brackets. Thus, the three permissible forms are:

From: mark@cbosgd.ATT.COM
From: mark@cbosgd.ATT.COM (Mark Horton)
From: Mark Horton <mark@cbosgd.ATT.COM>

Full names may contain any printing ASCII characters from space through tilde, except that they may not contain "(" (left parenthesis), ")" (right parenthesis), "<" (left angle bracket), or ">" (right angle bracket). Additional restrictions may be placed on full names by the mail standard, in particular, the characters "," (comma), ":" (colon), "@" (at), "!" (bang), "/" (slash), "=" (equal), and ";" (semicolon) are inadvisable in full names.

2.1.2. Date

The "Date" line (formerly "Posted") is the date that the message was originally posted to the network. Its format must be acceptable both in RFC-822 and to the getdate(3) routine that is provided with the Usenet software. This date remains unchanged as the message is propagated throughout the network. One format that is acceptable to both is:

Wdy, DD Mon YY HH:MM:SS TIMEZONE

Several examples of valid dates appear in the sample message above. Note in particular that ctime(3) format:

Wdy Mon DD HH:MM:SS YYYY

is not acceptable because it is not a valid RFC-822 date. However, since older software still generates this format, news implementations are encouraged to accept this format and translate it into an acceptable format.

There is no hope of having a complete list of timezones. Universal Time (GMT), the North American timezones (PST, PDT, MST, MDT, CST, CDT, EST, EDT) and the +/-hhmm offset specifed in RFC-822 should be supported. It is recommended that times in message headers be transmitted in GMT and displayed in the local time zone.

2.1.3. Newsgroups

The "Newsgroups" line specifies the newsgroup or newsgroups in which the message belongs. Multiple newsgroups may be specified, separated by a comma. Newsgroups specified must all be the names of existing newsgroups, as no new newsgroups will be created by simply posting to them.

Wildcards (e.g., the word "all") are never allowed in a "Newsgroups" line. For example, a newsgroup comp.all is illegal, although a newsgroup rec.sport.football is permitted.

If a message is received with a "Newsgroups" line listing some valid newsgroups and some invalid newsgroups, a host should not remove invalid newsgroups from the list. Instead, the invalid newsgroups should be ignored. For example, suppose host A subscribes to the classes btl.all and comp.all, and exchanges news messages with host B, which subscribes to comp.all but not btl.all. Suppose A receives a message with Newsgroups: comp.unix,btl.general.

This message is passed on to B because B receives comp.unix, but B does not receive btl.general. A must leave the "Newsgroups" line unchanged. If it were to remove btl.general, the edited header could eventually re-enter the btl.all class, resulting in a message that is not shown to users subscribing to btl.general. Also, follow-ups from outside btl.all would not be shown to such users.

2.1.4. Subject

The "Subject" line (formerly "Title") tells what the message is about. It should be suggestive enough of the contents of the message to enable a reader to make a decision whether to read the message based on the subject alone. If the message is submitted in response to another message (e.g., is a follow-up) the default subject should begin with the four characters "Re:", and the "References" line is required. For follow-ups, the use of the "Summary" line is encouraged.

2.1.5. Message-ID

The "Message-ID" line gives the message a unique identifier. The Message-ID may not be reused during the lifetime of any previous message with the same Message-ID. (It is recommended that no Message-ID be reused for at least two years.) Message-ID's have the syntax:

<string not containing blank or ">">

In order to conform to RFC-822, the Message-ID must have the format:

<unique@full_domain_name>

where full_domain_name is the full name of the host at which the message entered the network, including a domain that host is in, and unique is any string of printing ASCII characters, not including "<" (left angle bracket), ">" (right angle bracket), or "@" (at sign).

For example, the unique part could be an integer representing a sequence number for messages submitted to the network, or a short string derived from the date and time the message was created. For example, a valid Message-ID for a message submitted from host ucbvax in domain "Berkeley.EDU" would be "<4123@ucbvax.Berkeley.EDU>". Programmers are urged not to make assumptions about the content of Message-ID fields from other hosts, but to treat them as unknown character strings. It is not safe, for example, to assume that a Message-ID will be under 14 characters, that it is unique in the first 14 characters, nor that is does not contain a "/".

The angle brackets are considered part of the Message-ID. Thus, in references to the Message-ID, such as the ihave/sendme and cancel control messages, the angle brackets are included. White space characters (e.g., blank and tab) are not allowed in a Message-ID. Slashes ("/") are strongly discouraged. All characters between the angle brackets must be printing ASCII characters.

2.1.6. Path

This line shows the path the message took to reach the current system. When a system forwards the message, it should add its own name to the list of systems in the "Path" line. The names may be separated by any punctuation character or characters (except "." which is considered part of the hostname). Thus, the following are valid entries:

```
cbosgd!mhuxj!mhuxt
cbosgd, mhuxj, mhuxt
@cbosgd.ATT.COM,@mhuxj.ATT.COM,@mhuxt.ATT.COM
teklabs, zehntel, sri-unix@cca!decvax
```

(The latter path indicates a message that passed through decvax, cca, sri-unix, zehntel, and teklabs, in that order.) Additional names should be added from the left. For example, the most recently added name in the fourth example was teklabs. Letters, digits, periods and hyphens are considered part of host names; other punctuation, including blanks, are considered separators.

Normally, the rightmost name will be the name of the originating system. However, it is also permissible to include an extra entry on the right, which is the name of the sender. This is for upward compatibility with older systems.

The "Path" line is not used for replies, and should not be taken as a mailing address. It is intended to show the route the message traveled to reach the local host. There are several uses for this information. One is to monitor USENET routing for performance

reasons. Another is to establish a path to reach new hosts. Perhaps the most important use is to cut down on redundant USENET traffic by failing to forward a message to a host that is known to have already received it. In particular, when host A sends a message to host B, the "Path" line includes A, so that host B will not immediately send the message back to host A. The name each host uses to identify itself should be the same as the name by which its neighbors know it, in order to make this optimization possible.

A host adds its own name to the front of a path when it receives a message from another host. Thus, if a message with path "A!X!Y!Z" is passed from host A to host B, B will add its own name to the path when it receives the message from A, e.g., "B!A!X!Y!Z". If B then passes the message on to C, the message sent to C will contain the path "B!A!X!Y!Z", and when C receives it, C will change it to "C!B!A!X!Y!Z".

Special upward compatibility note: Since the "From", "Sender", and "Reply-To" lines are in Internet format, and since many USENET hosts do not yet have mailers capable of understanding Internet format, it would break the reply capability to completely sever the connection between the "Path" header and the reply function. It is recognized that the path is not always a valid reply string in older implementations, and no requirement to fix this problem is placed on implementations. However, the existing convention of placing the host name and an "!" at the front of the path, and of starting the path with the host name, an "!", and the user name, should be maintained when possible.

2.2. Optional Headers

2.2.1. Reply-To

This line has the same format as "From". If present, mailed replies to the author should be sent to the name given here. Otherwise, replies are mailed to the name on the "From" line. (This does not prevent additional copies from being sent to recipients named by the replier, or on "To" or "Cc" lines.) The full name may be optionally given, in parentheses, as in the "From" line.

2.2.2. Sender

This field is present only if the submitter manually enters a "From" line. It is intended to record the entity responsible for submitting the message to the network. It should be verified by the software at the submitting host.

For example, if John Smith is visiting CCA and wishes to post a message to the network, using friend Sarah Jones' account, the message might read:

```
From: smith@ucbvax.Berkeley.EDU (John Smith)
Sender: jones@cca.COM (Sarah Jones)
```

If a gateway program enters a mail message into the network at host unix.SRI.COM, the lines might read:

```
From: John.Doe@A.CS.CMU.EDU
Sender: network@unix.SRI.COM
```

The primary purpose of this field is to be able to track down messages to determine how they were entered into the network. The full name may be optionally given, in parentheses, as in the "From" line.

2.2.3. Followup-To

This line has the same format as "Newsgroups". If present, follow-up messages are to be posted to the newsgroup or newsgroups listed here. If this line is not present, follow-ups are posted to the newsgroup or newsgroups listed in the "Newsgroups" line.

If the keyword poster is present, follow-up messages are not permitted. The message should be mailed to the submitter of the message via mail.

2.2.4. Expires

This line, if present, is in a legal USENET date format. It specifies a suggested expiration date for the message. If not present, the local default expiration date is used. This field is intended to be used to clean up messages with a limited usefulness, or to keep important messages around for longer than usual. For example, a message announcing an upcoming seminar could have an expiration date the day after the seminar, since the message is not useful after the seminar is over. Since local hosts have local policies for expiration of news (depending on available disk space, for instance), users are discouraged from providing expiration dates for messages unless there is a natural expiration date associated with the topic. System software should almost never provide a default "Expires" line. Leave it out and allow local policies to be used unless there is a good reason not to.

2.2.5. References

This field lists the Message-ID's of any messages prompting the submission of this message. It is required for all follow-up messages, and forbidden when a new subject is raised. Implementations should provide a follow-up command, which allows a user to post a follow-up message. This command should generate a "Subject" line which is the same as the original message, except that if the original subject does not begin with "Re:" or "re:", the four characters "Re:" are inserted before the subject. If there is no "References" line on the original header, the "References" line should contain the Message-ID of the original message (including the angle brackets). If the original message does have a "References" line, the follow-up message should have a "References" line containing the text of the original "References" line, a blank, and the Message-ID of the original message.

The purpose of the "References" header is to allow messages to be grouped into conversations by the user interface program. This allows conversations within a newsgroup to be kept together, and potentially users might shut off entire conversations without unsubscribing to a newsgroup. User interfaces need not make use of this header, but all automatically generated follow-ups should generate the "References" line for the benefit of systems that do use it, and manually generated follow-ups (e.g., typed in well after the original message has been printed by the machine) should be encouraged to include them as well.

It is permissible to not include the entire previous "References" line if it is too long. An attempt should be made to include a reasonable number of backwards references.

2.2.6. Control

If a message contains a "Control" line, the message is a control message. Control messages are used for communication among USENET host machines, not to be read by users. Control messages are distributed by the same newsgroup mechanism as ordinary messages. The body of the "Control" header line is the message to the host.

For upward compatibility, messages that match the newsgroup pattern "all.all.ctl" should also be interpreted as control messages. If no "Control" header is present on such messages, the subject is used as the control message. However, messages on newsgroups matching this pattern do not conform to this standard.

Also for upward compatibility, if the first 4 characters of the "Subject:" line are "cmsg", the rest of the "Subject:" line should be interpreted as a control message.

2.2.7. Distribution

This line is used to alter the distribution scope of the message. It is a comma separated list similar to the "Newsgroups" line. User subscriptions are still controlled by "Newsgroups", but the message is sent to all systems subscribing to the newsgroups on the "Distribution" line in addition to the "Newsgroups" line. For the message to be transmitted, the receiving site must normally receive one of the specified newsgroups AND must receive one of the specified distributions. Thus, a message concerning a car for sale in New Jersey might have headers including:

```
Newsgroups: rec.auto,misc.forsale
Distribution: nj,ny
```

so that it would only go to persons subscribing to rec.auto or misc. for sale within New Jersey or New York. The intent of this header is to restrict the distribution of a newsgroup further, not to increase it. A local newsgroup, such as nj.crazy-eddie, will probably not be propagated by hosts outside New Jersey that do not show such a newsgroup as valid. A follow-up message should default to the same "Distribution" line as the original message, but the user can change it to a more limited one, or escalate the distribution if it was originally restricted and a more widely distributed reply is appropriate.

2.2.8. Organization

The text of this line is a short phrase describing the organization to which the sender belongs, or to which the machine belongs. The intent of this line is to help identify the person posting the message, since host names are often cryptic enough to make it hard to recognize the organization by the electronic address.

2.2.9. Keywords

A few well-selected keywords identifying the message should be on this line. This is used as an aid in determining if this message is interesting to the reader.

2.2.10. Summary

This line should contain a brief summary of the message. It is usually used as part of a follow-up to another message. Again, it

is very useful to the reader in determining whether to read the message.

2.2.11. Approved

This line is required for any message posted to a moderated newsgroup. It should be added by the moderator and consist of his mail address. It is also required with certain control messages.

2.2.12. Lines

This contains a count of the number of lines in the body of the message.

2.2.13. Xref

This line contains the name of the host (with domains omitted) and a white space separated list of colon-separated pairs of newsgroup names and message numbers. These are the newsgroups listed in the "Newsgroups" line and the corresponding message numbers from the spool directory.

This is only of value to the local system, so it should not be transmitted. For example, in:

```
Path: seismo!111-crg!111-lcc!pyramid!decwrl!reid
From: reid@decwrl.DEC.COM (Brian Reid)
Newsgroups: news.lists,news.groups
Subject: USENET READERSHIP SUMMARY REPORT FOR SEP 86
Message-ID: <5658@decwrl.DEC.COM>
Date: 1 Oct 86 11:26:15 GMT
Organization: DEC Western Research Laboratory
Lines: 441
Approved: reid@decwrl.UUCP
Xref: seismo news.lists:461 news.groups:6378
```

the "Xref" line shows that the message is message number 461 in the newsgroup news.lists, and message number 6378 in the newsgroup news.groups, on host seismo. This information may be used by certain user interfaces.

3. Control Messages

This section lists the control messages currently defined. The body of the "Control" header line is the control message. Messages are a sequence of zero or more words, separated by white space (blanks or tabs). The first word is the name of the control message, remaining words are parameters to the message. The remainder of the header

and the body of the message are also potential parameters; for example, the "From" line might suggest an address to which a response is to be mailed.

Implementors and administrators may choose to allow control messages to be carried out automatically, or to queue them for annual processing. However, manually processed messages should be dealt with promptly.

Failed control messages should NOT be mailed to the originator of the message, but to the local "usenet" account.

3.1. Cancel

cancel <Message-ID>

If a message with the given Message-ID is present on the local system, the message is cancelled. This mechanism allows a user to cancel a message after the message has been distributed over the network.

If the system is unable to cancel the message as requested, it should not forward the cancellation request to its neighbor systems.

Only the author of the message or the local news administrator is allowed to send this message. The verified sender of a message is the "Sender" line, or if no "Sender" line is present, the "From" line. The verified sender of the cancel message must be the same as either the "Sender" or "From" field of the original message. A verified sender in the cancel message is allowed to match an unverified "From" in the original message.

3.2. Ihave/Sendme

ihave <Message-ID list> [<remotesys>]
sendme <Message-ID list> [<remotesys>]

This message is part of the ihave/sendme protocol, which allows one host (say A) to tell another host (B) that a particular message has been received on A. Suppose that host A receives message "<1234@ucbvax.Berkeley.edu>", and wishes to transmit the message to host B.

A sends the control message "ihave <1234@ucbvax.Berkeley.edu> A" to host B (by posting it to newsgroup to.B). B responds with the control message "sendme <1234@ucbvax.Berkeley.edu> B" (on newsgroup to.A), if it has not already received the message. Upon receiving

the sendme message, A sends the message to B.

This protocol can be used to cut down on redundant traffic between hosts. It is optional and should be used only if the particular situation makes it worthwhile. Frequently, the outcome is that, since most original messages are short, and since there is a high overhead to start sending a new message with UUCP, it costs as much to send the ihave as it would cost to send the message itself.

One possible solution to this overhead problem is to batch requests. Several Message-ID's may be announced or requested in one message. If no Message-ID's are listed in the control message, the body of the message should be scanned for Message-ID's, one per line.

3.3. Newgroup

newgroup <groupname> [moderated]

This control message creates a new newsgroup with the given name. Since no messages may be posted or forwarded until a newsgroup is created, this message is required before a newsgroup can be used. The body of the message is expected to be a short paragraph describing the intended use of the newsgroup.

If the second argument is present and it is the keyword moderated, the group should be created moderated instead of the default of unmoderated. The newgroup message should be ignored unless there is an "Approved" line in the same message header.

3.4. Rmgroup

rmgroup <groupname>

This message removes a newsgroup with the given name. Since the newsgroup is removed from every host on the network, this command should be used carefully by a responsible administrator. The rmgroup message should be ignored unless there is an "Approved:" line in the same message header.

3.5. Sendsys

sendsys (no arguments)

The sys file, listing all neighbors and the newsgroups to be sent to each neighbor, will be mailed to the author of the control message ("Reply-To", if present, otherwise "From"). This information is considered public information, and it is a requirement of membership in USENET that this information be provided on request, either automatically in response to this control message, or manually, by mailing the requested information to the author of the message. This information is used to keep the map of USENET up to date, and to determine where netnews is sent.

The format of the file mailed back to the author should be the same as that of the sys file. This format has one line per neighboring host (plus one line for the local host), containing four colon separated fields. The first field has the host name of the neighbor, the second field has a newsgroup pattern describing the newsgroups sent to the neighbor. The third and fourth fields are not defined by this standard. The sys file is not the same as the UUCP L.sys file. A sample response is:

```
From: cbosgd!mark  (Mark Horton)
Date: Sun, 27 Mar 83 20:39:37 -0500
Subject: response to your sendsys request
To: mark@cbosgd.ATT.COM

Responding-System: cbosgd.ATT.COM
cbosgd:osg,cb,btl,bell,world,comp,sci,rec,talk,misc,news,soc,to,
        test
ucbvax:world,comp,to.ucbvax:L:
cbosg:world,comp,bell,btl,cb,osg,to.cbosg:F:/usr/spool/outnews
        /cbosg
cbosgb:osg,to.cbosgb:F:/usr/spool/outnews/cbosgb
sescent:world,comp,bell,btl,cb,to.sescent:F:/usr/spool/outnews
        /sescent
npois:world,comp,bell,btl,ug,to.npois:F:/usr/spool/outnews/npois
mhuxi:world,comp,bell,btl,ug,to.mhuxi:F:/usr/spool/outnews/mhuxi
```

3.6. Version

version (no arguments)

The name and version of the software running on the local system is to be mailed back to the author of the message ("Reply-to" if present, otherwise "From").

3.7. Checkgroups

The message body is a list of "official" newsgroups and their description, one group per line. They are compared against the list of active newsgroups on the current host. The names of any obsolete or new newsgroups are mailed to the user "usenet" and descriptions of the new newsgroups are added to the help file used when posting news.

4. Transmission Methods

USENET is not a physical network, but rather a logical network resting on top of several existing physical networks. These networks include, but are not limited to, UUCP, the Internet, an Ethernet, the BLICN network, an NSC Hyperchannel, and a BERKNET. What is important is that two neighboring systems on USENET have some method to get a new message, in the format listed here, from one system to the other, and once on the receiving system, processed by the netnews software on that system. (On UNIX systems, this usually means the rnews program being run with the message on the standard input. <1>)

It is not a requirement that USENET hosts have mail systems capable of understanding the Internet mail syntax, but it is strongly recommended. Since "From", "Reply-To", and "Sender" lines use the Internet syntax, replies will be difficult or impossible without an Internet mailer. A host without an Internet mailer can attempt to use the "Path" header line for replies, but this field is not guaranteed to be a working path for replies. In any event, any host generating or forwarding news messages must have an Internet address that allows them to receive mail from hosts with Internet mailers, and they must include their Internet address on their From line.

4.1. Remote Execution

Some networks permit direct remote command execution. On these networks, news may be forwarded by spooling the rnews command with the message on the standard input. For example, if the remote system is called remote, news would be sent over a UUCP link with the command:

```
uux - remote!rnews
```

and on a Berknet:

```
net -mremote rnews
```

It is important that the message be sent via a reliable mechanism, normally involving the possibility of spooling, rather than direct real-time remote execution. This is because, if the remote system is down, a direct execution command will fail, and the message will never be delivered. If the message is spooled, it will eventually be delivered when both systems are up.

4.2. Transfer by Mail

On some systems, direct remote spooled execution is not possible. However, most systems support electronic mail, and a news message can be sent as mail. One approach is to send a mail message which is identical to the news message: the mail headers are the news headers, and the mail body is the news body. By convention, this mail is sent to the user newsmail on the remote machine.

One problem with this method is that it may not be possible to convince the mail system that the "From" line of the message is valid, since the mail message was generated by a program on a system different from the source of the news message. Another problem is that error messages caused by the mail transmission would be sent to the originator of the news message, who has no control over news transmission between two cooperating hosts and does not know whom to contact. Transmission error messages should be directed to a responsible contact person on the sending machine.

A solution to this problem is to encapsulate the news message into a mail message, such that the entire message (headers and body) are part of the body of the mail message. The convention here is that such mail is sent to user rnews on the remote system. A mail message body is generated by prepending the letter N to each line of the news message, and then attaching whatever mail headers are convenient to generate. The N's are attached to prevent any special lines in the news message from interfering with mail transmission, and to prevent any extra lines inserted by the mailer (headers, blank lines, etc.) from becoming part of the news message. A program on the receiving machine receives mail to rnews, extracting the message itself and invoking the rnews program. An example in this format might look like this:

```
Date: Mon, 3 Jan 83 08:33:47 MST
From: news@cbosgd.ATT.COM
Subject: network news message
To: rnews@npois.ATT.COM

NPath: cbosgd!mhuxj!harpo!utah-cs!sask!derek
NFrom: derek@sask.UUCP (Derek Andrew)
NNewsgroups: misc.test
NSubject: necessary test
NMessage-ID: <176@sask.UUCP>
NDate: Mon, 3 Jan 83 00:59:15 MST
N
NThis really is a test.  If anyone out there more than 6
Nhops away would kindly confirm this note I would
Nappreciate it.  We suspect that our news postings
Nare not getting out into the world.
N
```

Using mail solves the spooling problem, since mail must always be spooled if the destination host is down. However, it adds more overhead to the transmission process (to encapsulate and extract the message) and makes it harder for software to give different priorities to news and mail.

4.3. Batching

Since news messages are usually short, and since a large number of messages are often sent between two hosts in a day, it may make sense to batch news messages. Several messages can be combined into one large message, using conventions agreed upon in advance by the two hosts. One such batching scheme is described here; its use is highly recommended.

News messages are combined into a script, separated by a header of the form:

```
#! rnews 1234
```

where 1234 is the length of the message in bytes. Each such line is followed by a message containing the given number of bytes. (The newline at the end of each line of the message is counted as one byte, for purposes of this count, even if it is stored as <CARRIAGE RETURN><LINE FEED>.) For example, a batch of message might look like this:

```
#! rnews 239
From: jerry@eagle.ATT.COM (Jerry Schwarz)
Path: cbosgd!mhuxj!mhuxt!eagle!jerry
Newsgroups: news.announce
Subject: Usenet Etiquette -- Please Read
Message-ID: <642@eagle.ATT.COM>
Date: Fri, 19 Nov 82 16:14:55 EST
Approved: mark@cbosgd.ATT.COM

Here is an important message about USENET Etiquette.
#! rnews 234
From: jerry@eagle.ATT.COM (Jerry Schwarz)
Path: cbosgd!mhuxj!mhuxt!eagle!jerry
Newsgroups: news.announce
Subject: Notes on Etiquette message
Message-ID: <643@eagle.ATT.COM>
Date: Fri, 19 Nov 82 17:24:12 EST
Approved: mark@cbosgd.ATT.COM

There was something I forgot to mention in the last
message.
```

Batched news is recognized because the first character in the message is #. The message is then passed to the unbatcher for interpretation.

The second argument (in this example rnews) determines which batching scheme is being used. Cooperating hosts may use whatever scheme is appropriate for them.

5. The News Propagation Algorithm

This section describes the overall scheme of USENET and the algorithm followed by hosts in propagating news to the entire logical network. Since all hosts are affected by incorrectly formatted messages and by propagation errors, it is important for the method to be standardized.

USENET is a directed graph. Each node in the graph is a host computer, and each arc in the graph is a transmission path from one host to another host. Each arc is labeled with a newsgroup pattern, specifying which newsgroup classes are forwarded along that link. Most arcs are bidirectional, that is, if host A sends a class of newsgroups to host B, then host B usually sends the same class of newsgroups to host A. This bidirectionality is not, however, required.

USENET is made up of many subnetworks. Each subnet has a name, such

as comp or btl. Each subnet is a connected graph, that is, a path exists from every node to every other node in the subnet. In addition, the entire graph is (theoretically) connected. (In practice, some political considerations have caused some hosts to be unable to post messages reaching the rest of the network.)

A message is posted on one machine to a list of newsgroups. That machine accepts it locally, then forwards it to all its neighbors that are interested in at least one of the newsgroups of the message. (Site A deems host B to be "interested" in a newsgroup if the newsgroup matches the pattern on the arc from A to B. This pattern is stored in a file on the A machine.) The hosts receiving the incoming message examine it to make sure they really want the message, accept it locally, and then in turn forward the message to all their interested neighbors. This process continues until the entire network has seen the message.

An important part of the algorithm is the prevention of loops. The above process would cause a message to loop along a cycle forever. In particular, when host A sends a message to host B, host B will send it back to host A, which will send it to host B, and so on. One solution to this is the history mechanism. Each host keeps track of all messages it has seen (by their Message-ID) and whenever a message comes in that it has already seen, the incoming message is discarded immediately. This solution is sufficient to prevent loops, but additional optimizations can be made to avoid sending messages to hosts that will simply throw them away.

One optimization is that a message should never be sent to a machine listed in the "Path" line of the header. When a machine name is in the "Path" line, the message is known to have passed through the machine. Another optimization is that, if the message originated on host A, then host A has already seen the message. Thus, if a message is posted to newsgroup misc.misc, it will match the pattern misc.all (where all is a metasymbol that matches any string), and will be forwarded to all hosts that subscribe to misc.all (as determined by what their neighbors send them). These hosts make up the misc subnetwork. A message posted to btl.general will reach all hosts receiving btl.all, but will not reach hosts that do not get btl.all. In effect, the messages reaches the btl subnetwork. A messages posted to newsgroups misc.misc,btl.general will reach all hosts subscribing to either of the two classes.

Notes

<1> UNIX is a registered trademark of AT&T.